J 4

```
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY            ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY            ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY            ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY            ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY            ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY            ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY         ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY         ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY         ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
```

FILEID**OBJEXECHK

K 6

```
  000000    BBBBBBBB            JJ  EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  HH      HH  KK      KK
  000000    BBBBBBBB            JJ  EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  HH      HH  KK      KK
00      00  BB      BB          JJ  EE          XX      XX  EE          CC          HH      HH  KK      KK
00      00  BB      BB          JJ  EE          XX      XX  EE          CC          HH      HH  KK      KK
00      00  BB      BB          JJ  EE            XX  XX    EE          CC          HH      HH  KK    KK
00      00  BB      BB          JJ  EE            XX  XX    EE          CC          HH      HH  KK    KK
00      00  BBBBBBBB            JJ  EEEEEEEE        XX      EEEEEEEE    CC          HHHHHHHHHH  KKKKKK
00      00  BBBBBBBB            JJ  EEEEEEEE        XX      EEEEEEEE    CC          HHHHHHHHHH  KKKKKK
00      00  BB      BB  JJ      JJ  EE            XX  XX    EE          CC          HH      HH  KK    KK
00      00  BB      BB  JJ      JJ  EE            XX  XX    EE          CC          HH      HH  KK    KK
00      00  BB      BB  JJ      JJ  EE          XX      XX  EE          CC          HH      HH  KK      KK  ....
00      00  BB      BB  JJ      JJ  EE          XX      XX  EE          CC          HH      HH  KK      KK  ....
  000000    BBBBBBBB      JJJJJJ    EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  HH      HH  KK      KK  ....
  000000    BBBBBBBB      JJJJJJ    EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  HH      HH  KK      KK  ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 
0 %title 'OBJEXECHK - General Checking Routines'
0         module objexechk(
1                         ident='V04-000') = begin
1 
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 
1 !++
1 ! Facility:     VAX/VMS Analyze Facility, General Checking Routines
1 !
1 ! Abstract:     This module provides general checking routines for the
1 !               ANALYZE/OBJECT and ANALYZE/IMAGE command image.
1 !
1 !
1 ! Environment:
1 !
1 ! Author: Paul C. Anagnostopoulos, Creation Date: 15 January 1980
1 !
1 ! Modified By:
1 !
1 !       V03-002 MCN0158         Maria del C. Nasr       22-Mar-1984
1 !               Add new parameter to ANL$CHECK_SYMBOL routine to indicate
1 !               maximum size of symbol.  Also, eliminate declaration for
1 !               local loop counter I.
1 !
1 !       V03-001 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
1 !               Change the message prefix to ANLOBJ$_ to ensure that
1 !               message symbols are unique across all ANALYZEs.  This
1 !               is necessitated by the new merged message files.
1 !--
```

```
;    55          0054   1 %sbttl 'Module Declarations'
;    56          0055   1 !
;    57          0056   1 ! Libraries and Requires:
;    58          0057   1 !
;    59          0058   1
;    60          0059   1 library 'starlet';
;    61          0060   1 require 'objexereq';
;    62          0496   1
;    63          0497   1 !
;    64          0498   1 ! Table of Contents:
;    65          0499   1 !
;    66          0500   1
;    67          0501   1 forward routine
;    68          0502   1         anl$check_symbol: novalue,
;    69          0503   1         anl$check_when: novalue,
;    70          0504   1         anl$check_flags: novalue;
;    71          0505   1
;    72          0506   1 !
;    73          0507   1 ! External References:
;    74          0508   1 !
;    75          0509   1
;    76          0510   1 external routine
;    77          0511   1         anl$format_error;
;    78          0512   1
;    79          0513   1 !
;    80          0514   1 ! Own Variables:
;    81          0515   1 !
```

```
1 %sbttl 'ANL$CHECK_SYMBOL - Check Validity of Symbol'
1 !++
1 ! Functional Description:
1 !         This routine is called to check the validity of a symbol, such
1 !         as a module name or a global name.
1 !
1 ! Formal Parameters:
1 !         symbol          The address of a descriptor of the symbol.
1 !         sym_size        Maximum size of symbol
1 !
1 ! Implicit Inputs:
1 !         global data
1 !
1 ! Implicit Outputs:
1 !         global data
1 !
1 ! Returned Value:
1 !         none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$check_symbol(symbol, sym_size): novalue = begin
2
2 bind
2         symbol_dsc = .symbol: descriptor;
2
2 bind
2         symbol_table = ch$transtable(
2                                         rep  32 of (false),
2                                         rep   1 of (true),        ! space
2                                         rep   3 of (false),
2                                         rep   1 of (true),        ! dollar sign
2                                         rep   9 of (false),
2                                         rep   1 of (true),        ! period
2                                         rep   1 of (false),
2                                         rep  10 of (true),        ! digits
2                                         rep   7 of (false),
2                                         rep  26 of (true),        ! upper-case letters
2                                         rep   4 of (false),
2                                         rep   1 of (true),        ! underscore
2                                         rep 160 of (false));
2
2 builtin
2         spanc;
2
2
2 ! First we check the length of the symbol.
2
2 if (.symbol_dsc[len] lssu 1) or (.symbol_dsc[len] gtru .sym_size) then
2         anl$format_error(anlobj$_badsymlen,.sym_size);
2
2 ! Now we verify that the symbol is composed of the correct character set.
2
2 if spanc(symbol_dsc[len],.symbol_dsc[ptr],symbol_table,%ref(%x'ff')) nequ 0 then
```

```
2          anl$format_error(anlobj$_badsymchar);
2
2 ! Finally, make sure the symbol does not start with a digit.
2
2 if (ch$rchar(.symbol_dsc[ptr]) gequ '0') and
2    (ch$rchar(.symbol_dsc[ptr]) lequ '9')          then
2          anl$format_error(anlobj$_badsym1st);
2
2 return;
2
1 end;
```

```
                                        .TITLE  OBJEXECHK OBJEXECHK - General Checking Routines
                                        .IDENT  \V04-000\

                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    00# 00000 P.AAA:    .BYTE   0[32]                                          ;
                    01  00020           .BYTE   1                                              ;
                    00# 00021           .BYTE   0[3]                                           ;
                    01  00024           .BYTE   1                                              ;
                    00# 00025           .BYTE   0[9]                                           ;
                    01  0002E           .BYTE   1                                              ;
                    00  0002F           .BYTE   0                                              ;
                    01# 00030           .BYTE   1[10]                                          ;
                    00# 0003A           .BYTE   0[7]                                           ;
                    01# 00041           .BYTE   1[26]                                          ;
                    00# 0005B           .BYTE   0[4]                                           ;
                    01  0005F           .BYTE   1                                              ;
                    00# 00060           .BYTE   0[160]                                         ;

                              SYMBOL_TABLE=        P.AAA
                                        .EXTRN  ANLOBJ$_OK, ANLOBJ$_ANYTHING
                                        .EXTRN  ANLOBJ$_DATATYPE
                                        .EXTRN  ANLOBJ$_ERRORCOUNT
                                        .EXTRN  ANLOBJ$_ERRORNONE
                                        .EXTRN  ANLOBJ$_ERRORS, ANLOBJ$_EXEFIXA
                                        .EXTRN  ANLOBJ$_EXEFIXAIMAGE
                                        .EXTRN  ANLOBJ$_EXEFIXALINE
                                        .EXTRN  ANLOBJ$_EXEFIXCOUNT
                                        .EXTRN  ANLOBJ$_EXEFIXEXTRA
                                        .EXTRN  ANLOBJ$_EXEFIXFIXED
                                        .EXTRN  ANLOBJ$_EXEFIXFLAGS
                                        .EXTRN  ANLOBJ$_EXEFIXG
                                        .EXTRN  ANLOBJ$_EXEFIXGIMAGE
                                        .EXTRN  ANLOBJ$_EXEFIXGLINE
                                        .EXTRN  ANLOBJ$_EXEFIXLIST
                                        .EXTRN  ANLOBJ$_EXEFIXNAME
                                        .EXTRN  ANLOBJ$_EXEFIXNAME0
                                        .EXTRN  ANLOBJ$_EXEFIXP
                                        .EXTRN  ANLOBJ$_EXEFIXPSECT
                                        .EXTRN  ANLOBJ$_EXEFIXUP
                                        .EXTRN  ANLOBJ$_EXEFIXUPNONE
                                        .EXTRN  ANLOBJ$_EXEGST, ANLOBJ$_EXEHDR
                                        .EXTRN  ANLOBJ$_EXEHDRACTIVE
                                        .EXTRN  ANLOBJ$_EXEHDRBLKCOUNT
```

```
        .EXTRN  ANLOBJ$_EXEHDRCHANCOUNT
        .EXTRN  ANLOBJ$_EXEHDRCHANDEF
        .EXTRN  ANLOBJ$_EXEHDRDECECO
        .EXTRN  ANLOBJ$_EXEHDRDMT
        .EXTRN  ANLOBJ$_EXEHDRDST
        .EXTRN  ANLOBJ$_EXEHDRFILEID
        .EXTRN  ANLOBJ$_EXEHDRFIXED
        .EXTRN  ANLOBJ$_EXEHDRFLAGS
        .EXTRN  ANLOBJ$_EXEHDRGBLIDENT
        .EXTRN  ANLOBJ$_EXEHDRGST
        .EXTRN  ANLOBJ$_EXEHDRIDENT
        .EXTRN  ANLOBJ$_EXEHDRIMAGEID
        .EXTRN  ANLOBJ$_EXEHDRISD
        .EXTRN  ANLOBJ$_EXEHDRISDBASE
        .EXTRN  ANLOBJ$_EXEHDRISDCOUNT
        .EXTRN  ANLOBJ$_EXEHDRISDFLAGS
        .EXTRN  ANLOBJ$_EXEHDRISDGBLNAM
        .EXTRN  ANLOBJ$_EXEHDRISDNUM
        .EXTRN  ANLOBJ$_EXEHDRISDPFCDEF
        .EXTRN  ANLOBJ$_EXEHDRISDPFCSIZ
        .EXTRN  ANLOBJ$_EXEHDRISDTYPE
        .EXTRN  ANLOBJ$_EXEHDRISDVBN
        .EXTRN  ANLOBJ$_EXEHDRLINKID
        .EXTRN  ANLOBJ$_EXEHDRMATCH
        .EXTRN  ANLOBJ$_EXEHDRNAME
        .EXTRN  ANLOBJ$_EXEHDRNOPATCH
        .EXTRN  ANLOBJ$_EXEHDRPAGECOUNT
        .EXTRN  ANLOBJ$_EXEHDRPAGEDEF
        .EXTRN  ANLOBJ$_EXEHDRPATCH
        .EXTRN  ANLOBJ$_EXEHDRPATCHDATE
        .EXTRN  ANLOBJ$_EXEHDRPRIV
        .EXTRN  ANLOBJ$_EXEHDRROPATCH
        .EXTRN  ANLOBJ$_EXEHDRRWPATCH
        .EXTRN  ANLOBJ$_EXEHDRSYMDBG
        .EXTRN  ANLOBJ$_EXEHDRSYSVER
        .EXTRN  ANLOBJ$_EXEHDRTEXTVBN
        .EXTRN  ANLOBJ$_EXEHDRTIME
        .EXTRN  ANLOBJ$_EXEHDRTYPEEXE
        .EXTRN  ANLOBJ$_EXEHDRTYPELIM
        .EXTRN  ANLOBJ$_EXEHDRUSERECO
        .EXTRN  ANLOBJ$_EXEHDRXFER1
        .EXTRN  ANLOBJ$_EXEHDRXFER2
        .EXTRN  ANLOBJ$_EXEHDRXFER3
        .EXTRN  ANLOBJ$_EXEHEADING
        .EXTRN  ANLOBJ$_EXEPATCH
        .EXTRN  ANLOBJ$_FLAG, ANLOBJ$_HEXDATA
        .EXTRN  ANLOBJ$_HEXHEADING1
        .EXTRN  ANLOBJ$_HEXHEADING2
        .EXTRN  ANLOBJ$_INDMSGSEC
        .EXTRN  ANLOBJ$_INTERACT
        .EXTRN  ANLOBJ$_MASK, ANLOBJ$_OBJCPRREC
        .EXTRN  ANLOBJ$_OBJDBGREC
        .EXTRN  ANLOBJ$_OBJENV, ANLOBJ$_OBJEOMFLAGS
        .EXTRN  ANLOBJ$_OBJEOMREC
        .EXTRN  ANLOBJ$_OBJEOMSEVABT
        .EXTRN  ANLOBJ$_OBJEOMSEVERR
        .EXTRN  ANLOBJ$_OBJEOMSEVIGN
```

```
                        .EXTRN  ANLOBJ$_OBJEOMSEVRES
                        .EXTRN  ANLOBJ$_OBJEOMSEVSUC
                        .EXTRN  ANLOBJ$_OBJEOMSEVWRN
                        .EXTRN  ANLOBJ$_OBJEOMWREC
                        .EXTRN  ANLOBJ$_OBJFADPASSMECH
                        .EXTRN  ANLOBJ$_OBJGSDENV
                        .EXTRN  ANLOBJ$_OBJGSDENVFLAGS
                        .EXTRN  ANLOBJ$_OBJGSDENVPAR
                        .EXTRN  ANLOBJ$_OBJGSDEPM
                        .EXTRN  ANLOBJ$_OBJGSDEPMW
                        .EXTRN  ANLOBJ$_OBJGSDIDC
                        .EXTRN  ANLOBJ$_OBJGSDIDCENT
                        .EXTRN  ANLOBJ$_OBJGSDIDCFLAGS
                        .EXTRN  ANLOBJ$_OBJGSDIDCMATCH
                        .EXTRN  ANLOBJ$_OBJGSDIDCOBJ
                        .EXTRN  ANLOBJ$_OBJGSDIDCVALA
                        .EXTRN  ANLOBJ$_OBJGSDIDCVALB
                        .EXTRN  ANLOBJ$_OBJGSDLEPM
                        .EXTRN  ANLOBJ$_OBJGSDLPRO
                        .EXTRN  ANLOBJ$_OBJGSDLSY
                        .EXTRN  ANLOBJ$_OBJGSDPRO
                        .EXTRN  ANLOBJ$_OBJGSDPROW
                        .EXTRN  ANLOBJ$_OBJGSDPSC
                        .EXTRN  ANLOBJ$_OBJGSDPSCALIGN
                        .EXTRN  ANLOBJ$_OBJGSDPSCALLOC
                        .EXTRN  ANLOBJ$_OBJGSDPSCBASE
                        .EXTRN  ANLOBJ$_OBJGSDPSCFLAGS
                        .EXTRN  ANLOBJ$_OBJGSDREC
                        .EXTRN  ANLOBJ$_OBJGSDSPSC
                        .EXTRN  ANLOBJ$_OBJGSDSYM
                        .EXTRN  ANLOBJ$_OBJGSDSYMW
                        .EXTRN  ANLOBJ$_OBJGTXREC
                        .EXTRN  ANLOBJ$_OBJHDRIGNREC
                        .EXTRN  ANLOBJ$_OBJHEADING
                        .EXTRN  ANLOBJ$_OBJLITINDEX
                        .EXTRN  ANLOBJ$_OBJLNKREC
                        .EXTRN  ANLOBJ$_OBJLNMREC
                        .EXTRN  ANLOBJ$_OBJMHDCREATE
                        .EXTRN  ANLOBJ$_OBJMHDNAME
                        .EXTRN  ANLOBJ$_OBJMHDPATCH
                        .EXTRN  ANLOBJ$_OBJMHDREC
                        .EXTRN  ANLOBJ$_OBJMHDRECSIZ
                        .EXTRN  ANLOBJ$_OBJMHDSTRLVL
                        .EXTRN  ANLOBJ$_OBJMHDVERSION
                        .EXTRN  ANLOBJ$_OBJMTCCORRECT
                        .EXTRN  ANLOBJ$_OBJMTCINPUT
                        .EXTRN  ANLOBJ$_OBJMTCNAME
                        .EXTRN  ANLOBJ$_OBJMTCREC
                        .EXTRN  ANLOBJ$_OBJMTCSEQNUM
                        .EXTRN  ANLOBJ$_OBJMTCUIC
                        .EXTRN  ANLOBJ$_OBJMTCVERSION
                        .EXTRN  ANLOBJ$_OBJMTCWHEN
                        .EXTRN  ANLOBJ$_OBJPROARGCOUNT
                        .EXTRN  ANLOBJ$_OBJPROARGNUM
                        .EXTRN  ANLOBJ$_OBJPSECT
                        .EXTRN  ANLOBJ$_OBJSRCREC
                        .EXTRN  ANLOBJ$_OBJSTATHEADING1
```

```
        .EXTRN  ANLOBJ$_OBJSTATHEADING2
        .EXTRN  ANLOBJ$_OBJSTATLINE
        .EXTRN  ANLOBJ$_OBJSTATTOTAL
        .EXTRN  ANLOBJ$_OBJSYMBOL
        .EXTRN  ANLOBJ$_OBJSYMFLAGS
        .EXTRN  ANLOBJ$_OBJTIRARGINDEX
        .EXTRN  ANLOBJ$_OBJTIRCMD
        .EXTRN  ANLOBJ$_OBJTIRCMDSTK
        .EXTRN  ANLOBJ$_OBJTBTREC
        .EXTRN  ANLOBJ$_OBJTIRREC
        .EXTRN  ANLOBJ$_OBJTIRSTOIM
        .EXTRN  ANLOBJ$_OBJTIRVIELD
        .EXTRN  ANLOBJ$_OBJTTLREC
        .EXTRN  ANLOBJ$_OBJVALUE
        .EXTRN  ANLOBJ$_OBJUVALUE
        .EXTRN  ANLOBJ$_PROTECTION
        .EXTRN  ANLOBJ$_SEVERITY
        .EXTRN  ANLOBJ$_TEXT, ANLOBJ$_TEXTHDR
        .EXTRN  ANLOBJ$_NOSUCHMOD
        .EXTRN  ANLOBJ$_BADDATE
        .EXTRN  ANLOBJ$_BADHDRBLKCOUNT
        .EXTRN  ANLOBJ$_BADSEVERITY
        .EXTRN  ANLOBJ$_BADSYM1ST
        .EXTRN  ANLOBJ$_BADSYMCHAR
        .EXTRN  ANLOBJ$_BADSYMLEN
        .EXTRN  ANLOBJ$_EXEBADFIXUPEND
        .EXTRN  ANLOBJ$_EXEBADFIXUPISD
        .EXTRN  ANLOBJ$_EXEBADFIXUPVBN
        .EXTRN  ANLOBJ$_EXEBADISDS1
        .EXTRN  ANLOBJ$_EXEBADISDTYPE
        .EXTRN  ANLOBJ$_EXEBADMATCH
        .EXTRN  ANLOBJ$_EXEBADPATCHLEN
        .EXTRN  ANLOBJ$_EXEBADOBJ
        .EXTRN  ANLOBJ$_EXEBADTYPE
        .EXTRN  ANLOBJ$_EXEBADXFERO
        .EXTRN  ANLOBJ$_EXEHDRISDLONG
        .EXTRN  ANLOBJ$_EXEHDRLONG
        .EXTRN  ANLOBJ$_EXEISDLENDZRO
        .EXTRN  ANLOBJ$_EXEISDLENGBL
        .EXTRN  ANLOBJ$_EXEISDLENPRIV
        .EXTRN  ANLOBJ$_EXENOTNATIVE
        .EXTRN  ANLOBJ$_EXTRABYTES
        .EXTRN  ANLOBJ$_FIELDFIT
        .EXTRN  ANLOBJ$_FLAGERROR
        .EXTRN  ANLOBJ$_NOTOK, ANLOBJ$_OBJBADIDCMATCH
        .EXTRN  ANLOBJ$_OBJBADNUM
        .EXTRN  ANLOBJ$_OBJBADPOP
        .EXTRN  ANLOBJ$_OBJBADPUSH
        .EXTRN  ANLOBJ$_OBJBADTYPE
        .EXTRN  ANLOBJ$_OBJBADVIELD
        .EXTRN  ANLOBJ$_OBJEOMBADSEV
        .EXTRN  ANLOBJ$_OBJEOMMISSING
        .EXTRN  ANLOBJ$_OBJFADBADAVC
        .EXTRN  ANLOBJ$_OBJFADBADRBC
        .EXTRN  ANLOBJ$_OBJGSDBADALIGN
        .EXTRN  ANLOBJ$_OBJGSDBADSUBTYP
        .EXTRN  ANLOBJ$_OBJHDRRES
```

```
                                                                  .EXTRN  ANLOBJ$_OBJMHDBADRECSIZ
                                                                  .EXTRN  ANLOBJ$_OBJMHDBADSTRLVL
                                                                  .EXTRN  ANLOBJ$_OBJMHDMISSING
                                                                  .EXTRN  ANLOBJ$_OBJNONTIRCMD
                                                                  .EXTRN  ANLOBJ$_OBJNOPSC
                                                                  .EXTRN  ANLOBJ$_OBJNULLREC
                                                                  .EXTRN  ANLOBJ$_OBJPOSPACE
                                                                  .EXTRN  ANLOBJ$_OBJPROMINMAX
                                                                  .EXTRN  ANLOBJ$_OBJPSCABSLEN
                                                                  .EXTRN  ANLOBJ$_OBJRECTOOBIG
                                                                  .EXTRN  ANLOBJ$_OBJTIRRES
                                                                  .EXTRN  ANLOBJ$_OBJUNDEFENV
                                                                  .EXTRN  ANLOBJ$_OBJUNDEFLIT
                                                                  .EXTRN  ANLOBJ$_OBJUNDEFPSC
                                                                  .EXTRN  ANALYZE$_FACILITY
                                                                  .EXTRN  ANL$FORMAT_ERROR

                                                                  .PSECT  $CODE$,NOWRT,2

                                                 003C 00000       .ENTRY  ANL$CHECK_SYMBOL, Save R2,R3,R4,R5               ; 0540
                                  55       0000G  CF   9E 00002       MOVAB   ANL$FORMAT_ERROR, R5                             ;
                                  54          04  AC   D0 00007       MOVL    SYMBOL, R4                                       ; 0543
                                                  64   B5 0000B       TSTW    (R4)                                             ; 0567
                                                  08   13 0000D       BEQL    1$                                               ;
        08    AC              64          10      00   ED 0000F       CMPZV   #0, #16, (R4), SYM_SIZE                          ;
                                                  0C   1B 00015       BLEQU   2$                                               ;
                                          08      AC   DD 00017 1$:   PUSHL   SYM_SIZE                                         ; 0568
                                   00000000G      8F   DD 0001A       PUSHL   #ANLOBJ$_BADSYMLEN                               ;
                                  65              02   FB 00020       CALLS   #2, ANL$FORMAT_ERROR                             ;
        FF    8F      0000'  CF       04  B4      64   2B 00023 2$:   SPANC   (R4), @4(R4), SYMBOL_TABLE, #255                 ; 0572
                                                  02   12 0002C       BNEQ    3$                                               ;
                                                  51   D4 0002E       CLRL    R1                                               ;
                                                  51   D5 00030 3$:   TSTL    R1                                               ;
                                                  09   13 00032       BEQL    4$                                               ;
                                   00000000G      8F   DD 00034       PUSHL   #ANLOBJ$_BADSYMCHAR                              ; 0573
                                  65              01   FB 0003A       CALLS   #1, ANL$FORMAT_ERROR                             ;
                                  30          04  B4   91 0003D 4$:   CMPB    @4(R4), #48                                      ; 0577
                                                  0F   1F 00041       BLSSU   5$                                               ;
                                  39          04  B4   91 00043       CMPB    @4(R4), #57                                      ; 0578
                                                  09   1A 00047       BGTRU   5$                                               ;
                                   00000000G      8F   DD 00049       PUSHL   #ANLOBJ$_BADSYM1ST                               ; 0579
                                  65              01   FB 0004F       CALLS   #1, ANL$FORMAT_ERROR                             ;
                                                       04 00052 5$:   RET                                                      ; 0583

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %sbttl 'ANL$CHECK_WHEN - Check Date/Time Field'
1 !++
1 ! Functional Description:
1 !       This routine is called to check the format of a date/time field.
1 !
1 ! Formal Parameters:
1 !       when            The address of a descriptor of the field.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$check_when(when): novalue = begin
2
2 bind
2       when_dsc = .when: descriptor;
2
2 local
2       when_ok: byte,
2       char: byte,
2       char_ok: byte;
2
2
2 ! First we check the length of the date field.
2
2 when_ok = .when_dsc[len] eqlu 17;
2
2 ! Now we scan each character of the date and make sure that it is valid.
2
3 incru i from 0 to minu(.when_dsc[len]-1,17-1) do (
3
3       char = ch$rchar(.when_dsc[ptr]+.i);
3
3       case .i from 0 to 16 of set
3       [0, 12]:        char_ok = (.char eqlu ' ') or
3                                 ((.char gequ '0') and (.char lequ '9'));
3       [ 1,
3         7 to 10,
3        13,
3        15 to 16]:     char_ok = (.char gequ '0') and (.char lequ '9');
3
3       [2, 6]:         char_ok = .char eqlu '-';
3
3       [3 to 5]:       char_ok = ((.char gequ 'A') and (.char lequ 'Z')) or
3                                 ((.char gequ 'a') and (.char lequ 'z'));
3
3       [11]:           char_ok = .char eqlu ' ';
;   209          0641  3
;   210          0642  3            [14]:          char_ok = .char eqlu ':';
;   211          0643  3            tes;
;   212          0644  3
;   213          0645  3            when_ok = .when_ok and .char_ok;
;   214          0646  2  );
;   215          0647  2
;   216          0648  2  ! If the date wasn't valid, print an error message.
;   217          0649  2
;   218          0650  2  if not .when_ok then
;   219          0651  2            anl$format_error(anlobj$_baddate);
;   220          0652  2
;   221          0653  2  return;
;   222          0654  2
;   223          0655  1  end;



                                          03FC 00000          .ENTRY   ANL$CHECK_WHEN, Save R2,R3,R4,R5,R6,R7,R8,- ; 0606
                                                                       R9                                           :
                         56       04   AC  D0 00002           MOVL     WHEN, R6                                     ; 0609
                                       50  D4 00006           CLRL     R0                                           ; 0619
                         11            66  B1 00008           CMPW     (R6), #17                                    :
                                       02  12 0000B           BNEQ     1$                                           :
                                       50  D6 0000D           INCL     R0                                           :
                         58            50  90 0000F   1$:     MOVB     R0, WHEN_OK                                  :
                         57            66  3C 00012           MOVZWL   (R6), R7                                     ; 0623
                                       57  D7 00015           DECL     R7                                           :
                         10            57  D1 00017           CMPL     R7, #16                                      :
                                       03  1B 0001A           BLEQU    2$                                           :
                         57            10  D0 0001C           MOVL     #16, R7                                      :
                                       52  D4 0001F   2$:     CLRL     I                                            :
                                     00BF  31 00021           BRW      24$                                          :
                         51     04 B642  90 00024   3$:     MOVB     @4(R6)[I], CHAR                              ; 0625
             10          00            52  CF 00029           CASEL    I, #0, #16                                   ; 0627
    0061     0058      003D          0022     0002D   4$:     .WORD    5$-4$,-                                      :
    003D     0058      0061          0061     00035                    8$-4$,-                                      :
    009B     003D      003D          003D     0003D                    11$-4$,-                                     :
    003D     00A2      003D          0022     00045                    12$-4$,-                                     :
                                     003D     0004D                    12$-4$,-                                     :
                                                                       12$-4$,-                                     :
                                                                       11$-4$,-                                     :
                                                                       8$-4$,-                                      :
                                                                       8$-4$,-                                      :
                                                                       8$-4$,-                                      :
                                                                       8$-4$,-                                      :
                                                                       18$-4$,-                                     :
                                                                       5$-4$,-                                      :
                                                                       8$-4$,-                                      :
                                                                       19$-4$,-                                     :
                                                                       8$-4$,-                                      :
                                                                       8$-4$                                        :
                                       54  D4 0004F   5$:     CLRL     R4                                           ; 0628
                         20            51  91 00051           CMPB     CHAR, #32                                    :
                                       02  12 00054           BNEQ     6$                                           :
```

```
                                  54  D6 00056          INCL    R4                         :
                                  53  D4 00058 6$:      CLRL    R3                         : 0629
                      30          51  91 0005A          CMPB    CHAR, #48                  :
                                  02  1F 0005D          BLSSU   7$                         :
                                  53  D6 0005F          INCL    R3                         :
                                  50  D4 00061 7$:      CLRL    R0                         :
                      39          51  91 00063          CMPB    CHAR, #57                  :
                                  52  1B 00066          BLEQU   16$                        :
                                  52  11 00068          BRB     17$                        :
                                  53  D4 0006A 8$:      CLRL    R3                         : 0633
                      30          51  91 0006C          CMPB    CHAR, #48                  :
                                  02  1F 0006F          BLSSU   9$                         :
                                  53  D6 00071          INCL    R3                         :
                                  50  D4 00073 9$:      CLRL    R0                         :
                      39          51  91 00075          CMPB    CHAR, #57                  :
                                  02  1A 00078          BGTRU   10$                        :
                                  50  D6 0007A          INCL    R0                         :
                      54          53  D2 0007C 10$:     MCOML   R3, R4                     :
          55          50          54  8B 0007F          BICB3   R4, R0, CHAR_OK            :
                                  56  11 00083          BRB     23$                        :
                                  50  D4 00085 11$:     CLRL    R0                         : 0635
                      2D          51  91 00087          CMPB    CHAR, #45                  :
                                  4A  13 0008A          BEQL    21$                        :
                                  4A  11 0008C          BRB     22$                        :
                                  50  D4 0008E 12$:     CLRL    R0                         : 0637
                 41   8F          51  91 00090          CMPB    CHAR, #65                  :
                                  02  1F 00094          BLSSU   13$                        :
                                  50  D6 00096          INCL    R0                         :
                                  54  D4 00098 13$:     CLRL    R4                         :
                 5A   8F          51  91 0009A          CMPB    CHAR, #90                  :
                                  02  1A 0009E          BGTRU   14$                        :
                                  54  D6 000A0          INCL    R4                         :
                      53          50  D2 000A2 14$:     MCOML   R0, R3                     :
                      54          53  CA 000A5          BICL2   R3, R4                     :
                                  53  D4 000A8          CLRL    R3                         : 0638
                 61   8F          51  91 000AA          CMPB    CHAR, #97                  :
                                  02  1F 000AE          BLSSU   15$                        :
                                  53  D6 000B0          INCL    R3                         :
                                  50  D4 000B2 15$:     CLRL    R0                         :
                 7A   8F          51  91 000B4          CMPB    CHAR, #122                 :
                                  02  1A 000B8          BGTRU   17$                        :
                                  50  D6 000BA 16$:     INCL    R0                         :
                      59          53  D2 000BC 17$:     MCOML   R3, R9                     :
                      50          59  CA 000BF          BICL2   R9, R0                     :
          55          50          54  89 000C2          BISB3   R4, R0, CHAR_OK            :
                                  13  11 000C6          BRB     23$                        : 0637
                                  50  D4 000C8 18$:     CLRL    R0                         : 0640
                      20          51  91 000CA          CMPB    CHAR, #32                  :
                                  05  11 000CD          BRB     20$                        :
                                  50  D4 000CF 19$:     CLRL    R0                         : 0642
                      3A          51  91 000D1          CMPB    CHAR, #58                  :
                                  02  12 000D4 20$:     BNEQ    22$                        :
                                  50  D6 000D6 21$:     INCL    R0                         :
                      55          50  90 000D8 22$:     MOVB    R0, CHAR_OK                :
                      50          55  92 000DB 23$:     MCOMB   CHAR_OK, R0                : 0645
                      58          50  8A 000DE          BICB2   R0, WHEN_OK                :
                                  52  D6 000E1          INCL    I                          : 0623
```

```
                        57                 52  D1 000E3 24$:    CMPL    1, R7                                   :
                                           03  1A 000E6         BGTRU   25$                                     :
                                         FF39  31 000E8         BRW     3$                                      :
                        DB                 58  E8 000EB 25$:    BLBS    WHEN_OK, 26$                            : 0650
                            00000000G      8F  DD 000EE         PUSHL   #ANLOBJ$_BADDATE                        : 0651
                   0000G  CF               01  FB 000F4         CALLS   #1, ANL$FORMAT_ERROR                    :
                                               04 000F9 26$:    RET                                             : 0655

; Routine Size:  250 bytes,    Routine Base:  $CODE$ + 0053
;  225       0656  1 %sbttl 'ANL$CHECK_FLAGS - Check Flag Usage'
;  226       0657  1 !++
;  227       0658  1 ! Functional Description:
;  228       0659  1 !       This routine is called to check the usage of flags in a flag
;  229       0660  1 !       byte/word/longword.
;  230       0661  1 !
;  231       0662  1 ! Formal Parameters:
;  232       0663  1 !       flags           A longword containing the flags to be checked.
;  233       0664  1 !       flag_def        A longword vector defining the valid flags.  The
;  234       0665  1 !                       zeroth longword contains the bit number of the
;  235       0666  1 !                       last valid flag.  The remaining longwords contain
;  236       0667  1 !                       zero if the flag is unused, non-zero otherwise.
;  237       0668  1 !
;  238       0669  1 ! Implicit Inputs:
;  239       0670  1 !       global data
;  240       0671  1 !
;  241       0672  1 ! Implicit Outputs:
;  242       0673  1 !       global data
;  243       0674  1 !
;  244       0675  1 ! Returned Value:
;  245       0676  1 !       none
;  246       0677  1 !
;  247       0678  1 ! Side Effects:
;  248       0679  1 !
;  249       0680  1 !--
;  250       0681  1
;  251       0682  1
;  252       0683  2 global routine anl$check_flags(flags,flag_def): novalue = begin
;  253       0684  2
;  254       0685  2 bind
;  255       0686  2         flags_vector = flags: bitvector[],
;  256       0687  2         flag_def_vector = .flag_def: vector[,long];
;  257       0688  2
;  258       0689  2
;  259       0690  2 ! We will simply sit in a loop scanning the flag bits.  If any flag is
;  260       0691  2 ! set but undefined, we will issue an error message.
;  261       0692  2
;  262       0693  3 incru i from 0 to 31 do (
;  263       0694  3         if .flags_vector[.i] then
;  264       0695  4                 if .i lequ .flag_def_vector[0] then (
;  265       0696  4                         if .flag_def_vector[.i+1] eqlu 0 then
;  266       0697  4                                 anl$format_error(anlobj$_flagerror,.i)
;  267       0698  3                 ) else
;  268       0699  3                         anl$format_error(anlobj$_flagerror,.i);
;  269       0700  2 );
;  270       0701  2
;  271       0702  2 return;
;  272       0703  2
;  273       0704  1 end;



                                              0004 00000        .ENTRY  ANL$CHECK_FLAGS, Save R2                 ; 0683
                                           52  D4 00002         CLRL    I                                        ; 0693
              1D        04   AC            52  E1 00004 1$:     BBC     I, FLAGS_VECTOR, 3$                      ; 0694
```

```
                          08   BC              52  D1 00009          CMPL    I, @FLAG_DEF                          ; 0695
                                               0A  1A 0000D          BGTRU   2$                                    ;
                          50          08 BC42  DE 0000F          MOVAL   @FLAG_DEF[I], R0                      ; 0696
                                         04    A0  D5 00014          TSTL    4(R0)                                 ;
                                               0D  12 00017          BNEQ    3$                                    ;
                                               52  DD 00019 2$:      PUSHL   I                                     ; 0699
                                    00000000G  8F  DD 0001B          PUSHL   #ANLOBJ$_FLAGERROR                    ;
                       0000G   CF              02  FB 00021          CALLS   #2, ANL$FORMAT_ERROR                  ;
                                               52  D6 00026 3$:      INCL    I                                     ; 0693
                               1F              52  D1 00028          CMPL    I, #31                                ;
                                               D7  1B 0002B          BLEQU   1$                                    ;
                                                   04 0002D          RET                                           ; 0704

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 014D


;  274          0705  1
;  275          0706  0 end eludom




;                                      PSECT SUMMARY
;
;       Name                         Bytes                         Attributes
;
;  $PLIT$                              256  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                              379  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                             Library Statistics
;
;                                          -------- Symbols --------       Pages        Processing
;       File                               Total    Loaded   Percent       Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        10         0         581          00:01.0




;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OBJEXECHK/OBJ=OBJ$:OBJEXECHK MSRC$:OBJEXECHK/UPDATE=(ENH$:OBJEXECHK)

; Size:           379 code + 256 data bytes
; Run Time:          00:10.1
; Elapsed Time:      00:21.1
; Lines/CPU Min:       4185
; Lexemes/CPU-Min:  13974
; Memory Used:   143 pages
; Compilation Complete
```

0006 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY